The George Washington University
WASHINGTON DC

OFFICE OF SPONSORED RESEARCH

May 16, 1997

Charles K. Hayes
Administrative Contracting Officer
Department of Navy
Atlanta Regional Office
100 Alabama Street, N.W.
Suite 4R15
Atlanta, GA 30303-3104

Re: Contract No. N00014-94-K-2009

Dear Mr. Hayes:

On behalf of Principal Investigator James K. Hahn, The George Washington University is pleased to submit the final technical report for the above mentioned contract. Also, in order to close out this award, The University is submitting the following documents:

a) final report of inventions and subcontracts --DD Form 882
b) final DOD property report --DD Form 1662

If you have any questions, please call Gianna Rudolph, Post Award Coordinator, at (202) 994-6257. Thank you.

Sincerely,

Helen Spencer
Director

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

DTIC QUALITY INSPECTED 4

enclosures

cc: Nick Kaplan
James Clements
Director, NRL
Defense Technical Information Center

2121 EYE STREET, N.W. • WASHINGTON, DC 20052 • (202) 994-6255 • FAX (202) 994-9137

A Rendering System for Photo-Realistic Images and Virtual Environments: Final Report

James K. Hahn
Director: Laboratory for Advanced Applications in Medicine
Director: Computer Graphics Laboratory

Department of Electrical Engineering and Computer Science
The George Washington University
Washington, DC 20052

1. Introduction

The initial proposal was for a joint project with Prof. Ihm Insung at Sogang University for developing a new rendering system. The rendering system was to be based on our ongoing project for a photo-realistic implementation of RenderMan Interface proposed by Pixar. In the first phase, we explored the requirements of such a renderer and how the present system needs to be changed to meet those requirements. This formed the basis for a photo-realistic rendering system that incorporates global illumination. In the second phase, we were to extend the system to real-time rendering. In particular, we are interested in real-time rendering of medical data. We wanted to include the traditional volumetric techniques as well as novel approaches in the use of texture mapping hardware and data segmentation and registration techniques for feature extraction and polygonal rendering. The system was to be used as the rendering part of a virtual environment system for surgical simulation.

We have already transferred our RenderMan interface-based rendering system to Sogang University. We have been working with Sogang University on a consultation basis for the first phase of the project. The present report will concentrate on the second phase, where we explore the use of real-time rendering systems for medical simulation applications. A paper representing the content of this work has been accepted to be published in IEEE Visualization '96.

The renderer is a part of a prototype system for simulating catheter insertion for surgical procedures, such as those used in the treatment of brain aneurysms. The short-term goal of the project is to provide the surgeon with an environment resembling the one actually used in the surgical procedure. Long term, we hope to identify which cues and levels of fidelity are necessary to effectively simulate surgical procedures, and, more generally, to make a contribution to the study of virtual environment design principles.

2. Medical Background

The surgical treatment of intracranial aneurysms is complex. Intracranial aneurysms have a variable geometric size, shape and wall thickness. The parent artery of origin, the perforating vessels adherent to the aneurysm wall, and the branches of the brain artery need to be spared, whereas the aneurysm has to be completely excluded normally by the clipping technique. Neighboring structures such as the cranial nerves, bone, and dura mater, as well as the brain which must be displaced by retraction, make the operation more challenging.

A normal aneurysm surgery has several steps. Initially, the aneurysm is exposed by craniotomy and the displacement of brain. Sometimes, the additional removal of bone at the base of the skull is also necessary. The aneurysm is then dissected from the surrounding structures. Frequently, the pressure in the aneurysm is lowered during the dissection by temporarily clipping the parent vessel proximal to (and sometimes also distal to) the aneurysm. Following dissection, the surgeon applies various types of clips singly or multiply across the neck of the aneurysm to exclude it from the parent vessels. Many types of clips are available, varying in length, shape, and closing force. Various types of clip appliers also are available.

Occasionally, it may not be possible to clip an aneurysm. Reasons for this include the origination of important vessels from the aneurysm wall, marked thickening of the aneurysm wall or the wall of the parent artery due to atherosclerosis, the presence of a marked degree of thrombosis inside the aneurysm, and a relatively inaccessible location of the aneurysm. In such cases, treatment may include a vein graft bypass of the aneurysm and the parent artery,

or the placement of fine metal coils into the aneurysm. The purpose of this simulator is to replicate the metal coil placement procedure.

Aneurysm surgery can be complicated by the rupture of the aneurysm during dissection, occlusion of perforating vessel, the parent artery, or its branches, and inadequate clipping of the aneurysm. Other complications may be caused by injury to the brain or the surrounding cranial nerves.

The training of aneurysm surgeons is a long and arduous task, limited by the number of operations at any one center. The proper selection of treatment involves much experience, a three-dimensional appreciation of the anatomy at operation, and often a process of trial and error in the application of clips.

If aneurysm surgery could be simulated in virtual reality, it would:

1) Help in the training of surgeons,

2) Provide for making better choices among the types of treatment (clipping, venous bypass, or intravascular surgery) and in the case of clipping, to allow the proper selection of clips.

3) Improve the outcome of aneurysm operations in real patients, and

4) Serve as a model for virtual reality simulations of other intracranial operations.

2.1. Cathether Insertion

In cases where the aneurysm cannot be clipped, fine metal coils are placed into the aneurysm to restrict the supply of blood. The procedure involves the insertion of a very fine catheter into the vasculature starting from the femeral artery. The catheter is guided through the vasculature by the surgeon using his hand. Bifurcations in the vessels are navigated by rotating catheter to manipulate the head into the correct passages. Once the head of the catheter reaches the aneurysm, the coil is placed into the aneurysm. The surgeon's view of the catheter and the surrounding tissue is provided by x-ray images enhanced by radioactive agents introduced into the bloodstream (fluoroscope camera).

2.2. Surgery Simulator

Producing an effective and accurate simulation of the procedure requires several parts. We must:

1. acquire medical data, typically Computed Tomography (CT) or Magnetic Resonance (MR) data,

2. extract the vasculature from this data, and store it in a way that allows for efficient rendering and simulation,

3. simulate the interaction of the catheter with the blood vessel,

4. represent the surrounding tissue and bone to provide the surgeon with the cues typically found in the actual procedure,

5. render the vasculature on a workstation in a manner that provides adequate realism for the simulation, and

6. provide the proper haptic feedback to the surgeon.

This report will focus on a new approach to parts 4 and 5 above although, the entire system is functional at this point. The motivation for our approach comes from studying catheter insertion procedures, both live and on video tape. The nature of the images viewed by the surgeons during the procedure suggests a number of simplifications that make them amenable to real-time manipulation on the computer screen.

Movement of the fluoroscope camera is limited to a few degrees of freedom, namely longitudinal axial, translation and rotation, and zooming. This allows us to pre-compute display information based on the limited views possible using these degrees of freedom. Also, because we are concentrating on catheter insertion procedures, we can assume that the surgeon's focus will be on the vasculature. This makes the use of approximations more plausible; the objects located at the focus of the procedure are rendered at a high level of detail, using surface-based approaches, and the surrounding area is approximated, while still maintaining adequate fidelity.

CT data is used to recover the original X-Ray value for a particular ray path. The surrounding material is of great importance to the visualization, as it provides the necessary cues that enable the surgeon to successfully relate the

current location of the catheter to landmarks in the body. Because there is a certain level of commonality from patient to patient, surgeons learn where certain major vessel bifurcations are in relation to bodily landmarks.

3. Previous Work

Many algorithms for rendering volume data have been developed [elvi92, lore87, levo88, west90]. Though volume rendering is a very useful visualization tool, it is often quite slow. Two approaches have been used to speed up volume rendering: special purpose hardware [cull93], and parallel implementations [sing94]. Interactive frame rates have recently been achieved in some implementations [lacr95]. Generally, however, the cost of the machines or their lack of general availability make these solutions less attractive.

Hemminger, Cullip, and North [hemm94] and Wilson, Gelder, and Wilhelms [wils94] both describe methods that use the 3D texture hardware of the Silicon Graphics RealityEngine. Cabral, Cam, and Foran [cabr94] implemented volume reconstruction and rendering based on the Radon transform. This algorithm also used the 3D texture hardware of the SGI RealityEngine. Guan and Lipes [guan94] implemented a volume renderer on the Kubota Denali that made use of 3D texture hardware. Fraser [fras95] reports on another technique for using the graphics hardware found in the SGI RealityEngine to perform fast volume rendering. The approach used is to place the entire volume into the RE's texture memory as a 3D texture. Rendering is accomplished by making several planes that are parallel to the image plane and letting the graphics hardware apply the texture to those planes. The values at the planes are then composited using the graphics hardware alpha channels. Using this technique, the author claims image generation rates of 0.1 seconds for a 256^3 volume into a 256^3 image.

This is a successful technique, however, there appear to be three problems with it. The process requires a large amount of texture memory. The largest amount that can be put into an Infinite Reality graphics system is 64MB, making volumes larger than untenable (assuming only one byte per voxel). It appears that if the user wants to change the opacity function, the volume needs to be reprocessed and then reloaded into the texture memory. It also appears that the process may be significantly undersampling the volume when it creates the

sample planes. We have attempted to address some of these problems in our approach.

4. Application Description

Figure 1 shows the system structure for our simulation. The CT images are used for two purposes. First, a segmentation process is used to extract the vasculature. The spatial and connectivity information is stored in the blood vessel data structure. The simulation module uses these data, along with data from the force feedback device, to perform the simulation. The force feedback device provides us with real-time updates of the catheter's position. In addition, it is designed to provide realistic haptic feedback to the surgeon during the simulated procedure.

Second, the CT images are used to perform X-Ray Casting to produce a cylindrical texture map, as described in detail in Section 6. The rendering process uses the blood vessel data structure to render the vasculature using a surface-based approach. The vasculature is embedded in a cylinder, which is wrapped with the cylindrical texture map. Finally, the catheter data structure is used to render the catheter.

5. X-Ray Casting

A unique method for representing the large amounts of volumetric data in a compact form is currently being developed. X-Ray Casting is a two-step process. The first step is performed off-line and is view-independent, while the second step is done in real-time and is view-dependent. The first step involves creating a body map, while the second step involves texturing geometry in the output scene with the computed body map, based on viewer location. This is similar to the way environment mapping wraps a texture around geometry [fole90].

One of the major issues we needed to solve in refining our approach is selecting the geometry onto which we project the cast rays. At first thought, this seems irrelevant, since the orthographic projection we are using would mean we could use just a rectangle as projection geometry. However, this approach will not work because it does not take into account the final direction of view. We need to

cast rays from a direction, or point, onto a geometry which can then be viewed from different points of view without re-casting.

Anatomically, some landmarks are in front of the vasculature, and some are behind it. Camera movement during the procedure is constrained, allowing only rotation about, and translation along, the longitudinal axis of the body. For these reasons, we use a cylindrical projection about the position of the vasculature in the CT data. The algorithm for constructing a cylindrical body map from sequential CT images follows, and is shown graphically in Figure2.

```
X-Ray Casting Algorithm:
Set a circle with CylCenter (0,0) and CylRadius r
for slices (sequential CT images)
        Load the slice into memory
        for angle (0-360)
                Cast a ray from CylCenter to edge of circle, i.e.
                for radius (0-CylRadius)
                        Get a sample (pixel value)
                        Calculate the attenuation factor (see below)
                        Set the cylinder surface pixel value
                end
        end
end
Create the output image
```

The samples from one slice can be thought of as making up one circle of data. If we stack the circles from all the slices, we produce a cylindrical representation of the volume, with the radius of the cylinder equal to CylRadius and the height of the cylinder equal to the number of slices. This map can then be 'unwrapped' to form a rectangular (distorted) texture map of the sampled volumetric data, as shown in Figure 3. The output image size for our data set is 360x322x(SweepAngle x NumberOfSlices). It is in RGB format with a depth of 1. Our tests were conducted using CT images from the National Library of Medicine's Visible Man data set [nlm]. The input size of each slice in our case is 512x512x12 in RAW format.

Each CT slice has a header file with a defined constant, called the Hounsfield number offset [cho93], the image size in pixels, and a pixel thickness for each pixel in the image. By using these, we apply the formula:

$$\mu_i = \left(\frac{(I_i - 1024)}{1024} * \mu_\omega \right) + \mu_\omega \qquad (1)$$

$$\frac{E_f}{E_i} = \exp\left(-\sum_{i=1}^{n} (\mu_i * t_{slice})\right) \qquad (2)$$

where, m_i is the attenuation number, $m_w$is the attenuation number of water (we use 0.02/mm), I_i is the intensity of the pixel, t_{slice} is the thickness of a pixel for the current slice, and E_f/E_i is the output/input energy, which constitutes the overall attenuation factor. The (I_i -1024) in Equation (1) is the Hounsfield number. Since this is an exponential function, the final projection pixel value can be computed by multiplying each term.

After the image is created, we wrap (texture map) this 2D image onto the surface of a cylinder with the proper transparency value. We choose the size of the cylinder to be sufficient to provide a parallax effect when rotated about the blood vessel. In other words, we can see through the front side of the body, represented by the textured cylinder, to the blood vessel inside the body, which partially occludes the rear side of the body.

As shown in Figure 4, we are approximating a ray cast through the volume with sample points on the surface of the cylinder. We approximate the value at point A by sampling the values along OA, and at point B by sampling along OB. A correct orthographic computation would sample the values along AB. For q=180 degrees, the sample points accurately represent the data along the cast rays, as these rays are normal to the projection plane. The amount of distortion of the data increases as q diverges from 180 degrees in either direction. The impact of this distortion is minimized because the surgeon's attention is focused on the vasculature where q will be close to 180 degrees.

6. Results

Once embedded in the body map, the resulting combination of pseudo-volumetric cues and the vasculature produces a rendered image very similar to that seen on the operating-room fluoroscope by the surgeon. Figure 5 shows one frame taken from an actual surgical procedure (patient's head). Figure 6 shows a snapshot of our visualization (patient's thoracic region with descending aorta), using the X-Ray Casting technique.

The fact that our approach uses texture maps to approximate volumetric data means that any hardware support for texture mapping provides a significant performance boost. We are using a two-processor Silicon Graphics Power Onyx RE2, with 4MB of texture memory for the visualization of our simulation. In addition, we are also testing on a Silicon Graphics Maximum Impact with 1MB texture memory. We have not yet performed any objective performance tests, but both machines provide interactive frame rates.

The X-Ray Casting software used to create the body map is written in C++, and outputs a texture map. We are using Open Inventor [wern94] for rendering the environment.

7. Conclusion

A method for approximating volumetric medical data for surgical simulation was presented. The approach takes advantage of available texture memory, and provides realism approaching the actual display used in the operating room.

This is a simple approximation of 3D volume rendering. To get a better image, we are experimenting with putting multiple concentric cylinders inside the volume. This should give us more accurate results, because as the number of cylinders increases, distortion decreases, and we approach a full volumetric representation.

In addition to using a cylindrical projection, we are also examining the use of a spherical projection for visualizing CT head data. In this part of the surgical procedure, there is a different set of degrees of freedom for camera movement, so a cylindrical projection approach cannot capture all the necessary views of the volume data.

Further work will help us determine the extent to which our approximations and assumptions are valid within this domain, as well as shed some light on the applicability of our approach to more general visualization applications.

8. References

[cabr94] Cabral, B. and Cam, N. and Foran, J., "Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware", Proceedings 1994 Symposium on Volume Visualization", 1994, IEEE Computer Society Press

[cho93] Cho, Z.H., "Fundamental of Medical Imaging", Wiley, 1993

[cull93] Cullip, T. J. and U. Newman, "Accelerating volume reconstruction with 3d texture hardware", Tech. Rep. TR93-027, University of North Carolina, Department of Computer Science, 1993, Chapel Hill, NC.

[elvi92] Elvins, T.T., A Survey of Algorithms for Volume Visualization, Computer Graphics, (August, 1992) 26, 3, pp.194-201.

[fole90] Foley, J.D. and A. van Dam and S. K. Feiner and J. F. Hughes, "Computer Graphics: Principles and Practices (2nd Edition) ", 1990, Addison Wesley.

[fras95] Fraser, R, "Interactive Volume Rendering Using Advanced Graphics Architectures" at http://www.sgi.com/Technology/volume/VolumeRendering.html. 1995.

[guan94] Guan, S.Y. and Lipes, R., "Innovative volume rendering using 3D texture mapping", vol. 2164, pp. 382-392, In Proceedings Medical Imaging 1994--Image Capture, Formatting, and Display, 1994, SPIE, February.

[hemm94] Hemminger, B.M. and Cullip, T.J. and North, M.J., "Interactive Visualization of 3D Medical Image Data", Tech. Rep. TR94-032, University of North Carolina, Department of Computer Science, Chapel Hill, NC., 1994.

[vish] Information on The Visible Human Project can be found at http://www.nlm.nih.gov/research/visible/visible_human.html

[lacr95] Lacroute, P., "Real-Time Volume Rendering on Shared Memory Multiprocessors Using the Shear-Warp Factorization", 1995 Parallel Rendering Symposiun, (OCtober, 1995), pp. 15-22, ACM SIGGRAPH.

[levo98] Levoy, M., "Display of Surfaces from Volume Data, IEEE Computer Graphics and Applications, May, 1998, pp 29-37.

[lore87] Lorensen, W.E. and Cline, H.E., "Marching cubes: A high resolution 3D surface construction algorithm", Computer Graphics, (July, 1987), vol.21, pp.163-170, ACM SIGGRAPH.

[sing] Singh, J.P. and Gupta, A. and Levoy, M, "Parallel Visualization Algorithns: Performance and Architectural Implications", IEEE Computer, July, 7,27, pp.45-55.

[wern94] Wernecke, Josie, "The Inventor Mentor", Addison Wesley, 1994

[west90] Westover, L., "Footprint evaluation for volume rendering" Computer Graphics, (August, 1990), vol.24, ACM SIGGRAPH, pp.367-376.

[wils94] Wilson, O. and Gelder, A.V. and Wilhelms, J., "Direct Volume Rendering via 3D Textures", Tech-Rep. UCSC-CRL-94-19, University of California, Santa Cruz, Department of Computer and Information Science, Santa Cruz, CA, 1994.